notes$_{and guides}$

Release 0.1.0

Mar 11, 2020

Contents:

Virtual Environments with *virtualenv*

1.1 *virtualenv*

virtualenv is a third party Python package for handling isolated virtual environments. It is more feature-rich than the standard library package *venv*.

1.2 *virtualenvwrapper-win*

virtualenvwrapper-win, is an extension package to `virtualenv`, which provides easy-to-use wrappers around the commands.

Most of the commands shown below are for `virtualenvwrapper-win`

Note: This package works only in the regular Windows Command Prompt, not in PowerShell. The `-win` suffix, which is necessary for the Windows version of the package.

1.3 First time setup

Install the two packages if they aren't already:

```
pip install virtualenv
```

```
pip install virtualenvwrapper-win
```

1.4 Creating a virtual environment

```
mkvirtualenv <new_env_name>
```

The new virtual environment is then created and automatically activated.

The active environment is shown in parenthesis in the left side of the prompt.

1.5 Pair environment with working directory

Pair an environment with the current working directory by

```
setprojectdir .
```

This pairs the active environment to the current working directory.

1.6 Deactivating an environment

To deactivate an environment that is currently active, simply close the prompt that has the environment open. The parenthesis on the left showing the active environment will have disappeared once a new prompt is opened.

Alternatively, if you want to continue working in the same prompt, deactivate directly by

```
deactivate
```

Obviously, *virtualenvwrapper-win* knows exactly what environment is active, and thus which one to deactivate.

1.7 Start working on an existing environment

To start working on a project that is tied to an existing environment

```
workon <env_name>
```

The current directory will be changed to the root directory of the project that is paired with the virtual environment.

1.8 Remove an environment

```
rmvirtualenv <env_name>
```

1.9 Useful commands

List all virtual environments

```
lsvirtualenv
```

See the packages installed in the active virtual environment

```
pip list
```

1.10 Note on using *virtualenv* in PowerShell

As mentioned above, PowerShell is **not** supported to *virtualenvwrapper-win*.

Although, *virtualenv* does support PowerShell. However, it has a more verbose syntax since it misses a wrapper package like *virtualenvwrapper* for Bash and *virtualenvwrapper-win* for Windows Command Prompt.

E.g. an environment can be activated by running the *activate.ps1* PowerShell script tied to it.

1.11 References

- Article: https://timmyreilly.azurewebsites.net/python-pip-virtualenv-installation-on-windows/

CHAPTER 2

Indices and tables

- genindex
- modindex
- search